SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Sketch-based Stroke Generation in Chinese Flower Painting

YANG LiJie[1], XU TianChen[2*], DU JiXiang[1] & WU EnHua[3,4]

[1]*Huaqiao University, Xiamen* 361021*, China;*
[2]*Hong Kong University of Science and Technology, Hong Kong* 999077*, China;*
[3]*University of Macau, Macao* 999078*, China;*
[4]*State Key Lab. CS, IOS/CAS, Beijing* 100190*, China*

Appendix A Introduction

Unlike other types of paintings like western paintings, most Chinese paintings in freehand employ brush strokes economically to describe some specific objects or scenes purposefully and meaningfully. Mainly due to their unique appearance, more and more people, even with no background in paintings, participate in producing their own Chinese paintings. Meanwhile, the production tool in Chinese art forms is adopted in almost every graphics application (e.g., Adobe Illustrator, Adobe Photoshop, CorelDRAW etc.). However, the painting production in high quality requires expert knowledge and skill, and the existing tools demand complex and accurate inputs in terms of drawing techniques, or just migrate painting style to the overall object without concerning stroke performance in local part, as shown in the second column of Figure A1.

Flower painting, which is one of the major categories of Chinese painting, represents the key artistic characteristics of oriental paintings. In other word, its typical usages of stroke and ink color contribute to unique appeal of Chinese painting, since the strokes may be straight or curved, hard or soft, thick or thin, pale or dark, and the ink colors may be dry or running, blending or diffusing. Therefore, complexity and variety of strokes and ink colors, as well as professional skills make the work of painting production complicated and challenging. This method has been applied by unskilled users as a handy tool to master drawing techniques, and to generate their paintings. It has also been used by artists as a reference to improve their professional skills.

In this work, we explore a novel painting production tool with an easy user interface. Guided by a real flower image, our drawing tool generates its Chinese style painting by sketch-based style migration at the level of brush stroke, as shown in the third column of Figure A1. It provides the following benefits.

- We devise sketch-based style migration to provide a guidance for the composition and layout of a painting, which are rather difficult part in freehand drawing.
- In order to simplify user interaction, this method corrects the roughly sketched lines, including stroke outlines over an input painting and basic path (a basis of a stroke to be produced) over an input photo, to capture the user's intent through an opposite-direction search (Section Appendix B.1).
- A best stroke is selected from the style candidates circled by the user on the input painting using an energy function. It formalizes the optimum matching between the basic path and a stroke of the input painting (Section Appendix B.2).
- We migrate the style of a best stroke to basic path by analyzing and formalizing style features in shape and ink color (Section Appendix B.3).

Our main contributions are as follows:

- Compared with previous filtering methods for art stylization, a novel style migration technique in our work better performs art features in brush strokes that form a given painting.
- This work introduces a sketch-based framework of stylization that automatically rectify the inaccurate lines sketched by the user over the input photo and painting, a principle to decide the stroke with the best style for the selected path, as well as the process of style mapping by outline estimation upon stroke widths and texture synthesis in stroke region.

Besides, the effectiveness of our painting generation tool has been evaluated with a user study, which shows that the stylistic strokes by our tool are more satisfied than those from a commercial manual tool.

* Corresponding author (email: tianchenx@outlook.com)

Figure A1 Stylization of flower photos. Given two photos, second column shows images in style of Japanese painting by Adobe Photoshop, and third column shows images in style of Chinese flower painting by our tool.

Appendix A.1 Related Work

As far as we know, there has been little work on our topic. But several related works have been explored into the following aspects. Generally, Non-Photorealistic Rendering (NPR) [1] outputs strokes in artistic styles to describe something specific as hand paints by various brush models.

Physically-based Model. In order to provide an intuitive and natural feeling when users manipulate a pen-like device, it simulates the physical processes in drawing or painting by modeling the physical properties of the brush, media, etc.. Among them, Curtis et al. [2] imitated various artistic effects of watercolor using a shallow-water fluid simulation. Chu and Tai [3] developed a real-time paint system for simulating ink dispersion in absorbent paper. Also Xu et al. [4] proposed a novel "e-brush" for Chinese calligraphy and painting using only four attributes to capture the essential features of the brush. Recently, Chu et al. [5] presented new techniques to solve the problems of excessive loss of color detail and computational inefficiencies which existed in the traditional 3D brush model. Lu et al. [6] present an interactive, data-driven painting system that uses scanned images of real natural media to synthesize both new strokes and complex stroke interactions, obviating the need for physical simulation. In digital painting software, like Adobe brush packages, ArtRage, and Corel Painter, users draw with a mouse or a digital pen as a brush in the real world. However, manipulating a virtual brush exactly and effortlessly is complex and difficult for the users without much painting expertise.

Image-based Model. Compared with physical model, it avoids the great computational and controlling complexity. Stroke-Based Rendering [7] approaches are widely applied to convert automatically real images into paintings by placing discrete elements such as paint strokes or stipples. For western painting simulation, Hertzmann et al. [8] used *Image Analogy* technique to process photo images into some artistic effects, the method however produces very poor results when dealing with Chinese paintings in the experiments done by Hertzmann, because Chinese paintings have different painting styles for different parts of the image, and their algorithm can't distinguish them. To improve the stroke placement in painterly rendering, stroke processes [9] was proposed to allow users to adjust styles easily by controlling some intuitive parameters. For eastern painting simulation, special artistic effects in Chinese landscape painting were imitated. For instance, Way et al. [10] synthesized rock textures of two main "TSUN" techniques. Then they also drew a tree from a 3D polygonal model using the information of 3D model's surface [11]. By collecting only a few brush stroke texture primitives, Yu et al. [12] synthesized typical textures of the mountains and fog in landscape painting as the hand-made artwork. Besides, based on an ink footprint model, Xu et al. [13] extracted classified strokes from a 3D model and attached typical strokes' textures (e.g. contouring, texturing, dyeing, and pointing) on its surface. In [14, 15], the ancient Chinese painting and calligraphy are animated by reproducing their drawing and writing processes with brush techniques and art features in shape and ink color. Xie et al. [16] developed an interactive sketch-based system for converting a real photo into Sumi-e painting with automatic trajectory estimation in their brush model. With the similar motivation, Bang et al. [17] take a single photo image as input and produce an oriental brushwork-like image automatically as a result, basically following the techniques in conventional oriental paintings. A novel real-time, automatic framework is designed to convert images into Chinese ink painting style by texture mapping and synthesis [18]. Most previous models about imitating eastern painting style mainly process a whole picture instead of brush strokes which can hardly convey the richness and variety of appearance in Chinese painting.

Appendix B Principles and Algorithms

Given a sample painting I^{S} and an objective photo I^{P}, the kernel of our tool lies on three technical components as follows that automatically stylize a real flower image to be a Chinese style painting under the guidance of user sketched lines. As shown in Figure B1, we migrate the artistic style of the given painting to the real image in left red box and generate its painting in right red box.

Curve Correction. First, the user needs to select the sample strokes from the given painting I^{S} as style patterns by roughly circling, and sketch a basic path over the input photo I^{P} as the guidance of an output stroke. We analyze and correct the sketched lines to follow user's intents using image features. Note that the sketched lines of basic paths do not need to trace the real contour rigorously, since Chinese freehand drawing prefer individual and abstract creation.

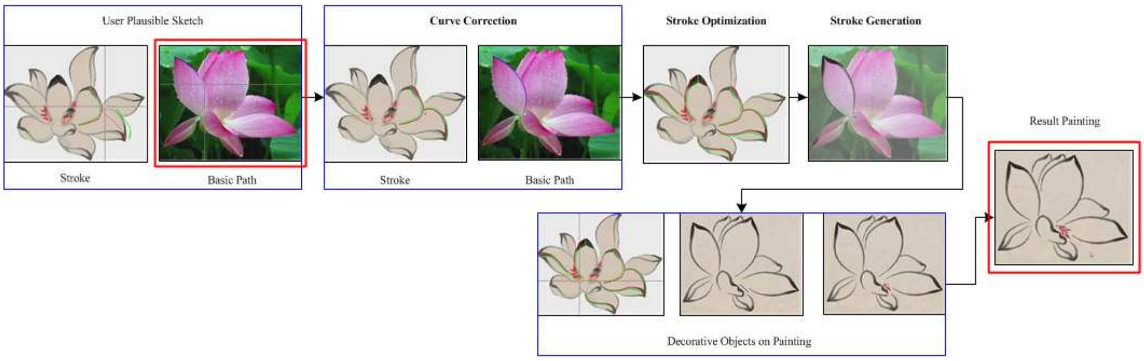


Figure B1 Overview of the proposed approach.

Stroke Optimization. Once the stroke candidates of style pattern are determined from the original painting, we select the optimal one for the basic path of a new stroke by formalizing the influence conditions of decision on candidate selection.

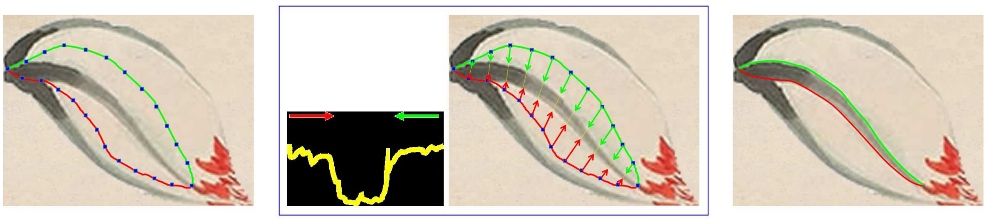
Stroke Generation. Finally, we map style features of the optimal stroke onto the basic path using stroke width calculation and stylistic texture synthesis.

For convenience, all math variables involved in the following content are summarized in Table **??** with brief descriptions (see Appendix).

Appendix B.1 Curve Correction

The goal of this step is to find the accurate outlines of one or more than one stroke candidates. In order to relax the requirement of user interaction, this tool allows the user to select sample strokes as the candidates of style pattern by imprecisely sketching their outlines over the given painting, and corrects the sketched lines by tightening them with an *opposite-directional search* technique.

Given two sides $\boldsymbol{L}^{\mathrm{S}}$ and $\boldsymbol{R}^{\mathrm{S}}$ of the sketched stroke outline, we fit them as two curves $\boldsymbol{L}^{\mathrm{S}}(t)$ and $\boldsymbol{R}^{\mathrm{S}}(t)$ by *Non-uniform rational B-spline (NURBS)* that can trace the sketched lines better than other fitting algorithms, and divide them into the same number of segments (Figure B2 (a)). Supposing that end points of each segment are regarded as feature points on $\boldsymbol{L}^{\mathrm{S}}(t)$ and $\boldsymbol{R}^{\mathrm{S}}(t)$, and the direction $\boldsymbol{\omega}(t)$ of a line from $\boldsymbol{L}^{\mathrm{S}}(t)$ to $\boldsymbol{R}^{\mathrm{S}}(t)$ is defined as the positive search direction, we relocate a pair of feature points at the positions of the global minima of $D_{+\omega}$ and $D_{-\omega}$, which are the directional derivatives of luminance values for each pixel on I^{S} along $\boldsymbol{\omega}(t)$ (red arrow) and $-\boldsymbol{\omega}(t)$ (green arrow) respectively (Figure B2 (b)).



(a) (b) (c)

Figure B2 Opposite-directional search in curve correction.

Furthermore, for some strokes intersected by other strokes, the *opposite-directional search* technique may bring about the concave or convex outline with wrong feature point in the area of intersection. As shown in Figure B3, taking the stroke circled by red and green lines for example, this technique find the feature point deviating from the real outline due to similar luminance values of two crossing strokes. For dealing with this exception, we process two steps iteratively: (1) detecting wrong feature point whose curvature is twice greater than the average curvature by calculating local curvature of each feature point on outline curve; and (2) correcting the location of wrong feature point by interpolation of its two neighboring feature points. However, some strokes can hardly be distinguished by human beings with weak edges such as the heads of three strokes in Figure B2. So we separate them by means of user-sketched lines.

As a result, a stroke in painting I^{S} is described by a triple-curve structure $(\boldsymbol{C}^{\mathrm{S}}, \boldsymbol{L}^{\mathrm{S}}, \boldsymbol{R}^{\mathrm{S}})$, where $\boldsymbol{C}^{\mathrm{S}}(t) \in I^{\mathrm{S}}$ is obtained by $\boldsymbol{C}^{\mathrm{S}}(t) = \left(\boldsymbol{L}^{\mathrm{S}}(t) + \boldsymbol{R}^{\mathrm{S}}(t)\right)/2$, and its domain F^{S} between $\boldsymbol{L}^{\mathrm{S}}$ and $\boldsymbol{R}^{\mathrm{S}}$ is called *sample-area*. Similarly, the stroke in photo I^{P} is marked by the triple curves $(\boldsymbol{C}^{\mathrm{P}}, \boldsymbol{L}^{\mathrm{P}}, \boldsymbol{R}^{\mathrm{P}})$. Besides, the paths $\boldsymbol{C}^{\mathrm{P}}$ over the given photo I^{P} are just smoothed by *NURBS*, since they are not necessarily consistent with real edges for the purpose of art creation in Chinese painting.

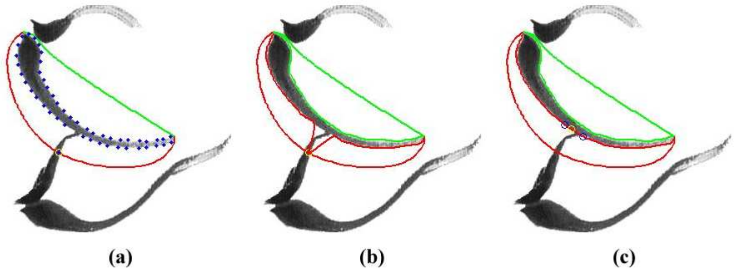


Figure B3 Exception case of *opposite-directional search* technique. (a) Feature point in yellow circle is found by mistake for stroke intersection. (b) Wrong point is detected by curvature calculation on its curve. (c) It is corrected by interpolation.

Appendix B.2 Stroke Optimization

The curve correction step will yield multiple stroke candidates with their precise outlines according to user sketch over the input painting, as long as the user selects more than one stroke. But determining the right stroke (i.e. style pattern) from them is somewhat puzzling, since Chinese painting emphasizes on variety and richness of stroke shape. Hence, our system proposes an energy function to automatically select the optimal one whose skeletal shape has the most similarity to the basic path on the input photo.

Supposing that there are n stroke candidates that are sketched and selected by the user, we call them as a candidate set including $4n$ elements $\mathcal{C} = \{\boldsymbol{S}_i^{\mathrm{S}} | i = 0, 1, ..., 4n-1\}$, which considers four cases of the strokes: current stroke $\boldsymbol{S}_{4j}^{\mathrm{S}}$, mirror stroke $\boldsymbol{S}_{4j+1}^{\mathrm{S}}$, reversed-path stroke $\boldsymbol{S}_{4j+2}^{\mathrm{S}}$ and mirror reversed-path stroke $\boldsymbol{S}_{4j+3}^{\mathrm{S}}$ $(j = 0, 1, ..., n-1)$. We defines them in terms of triple-curve structure as follows:

$$
\begin{cases}
\boldsymbol{S}_{4j}^{\mathrm{S}} = \left(\boldsymbol{C}_j^{\mathrm{S}}(t), \boldsymbol{L}_j^{\mathrm{S}}(t), \boldsymbol{R}_j^{\mathrm{S}}(t))\right), \\
\boldsymbol{S}_{4j+1}^{\mathrm{S}} = \left(\boldsymbol{C}_j^{\mathrm{S}}(t), \boldsymbol{R}_j^{\mathrm{S}}(t), \boldsymbol{L}_j^{\mathrm{S}}(t)\right), \\
\boldsymbol{S}_{4j+2}^{\mathrm{S}} = \left(\boldsymbol{C}_j^{\mathrm{S}}(1-t), \boldsymbol{L}_j^{\mathrm{S}}(1-t), \boldsymbol{R}_j^{\mathrm{S}}(1-t)\right), \\
\boldsymbol{S}_{4j+3}^{\mathrm{S}} = \left(\boldsymbol{C}_j^{\mathrm{S}}(1-t), \boldsymbol{R}_j^{\mathrm{S}}(1-t), \boldsymbol{L}_j^{\mathrm{S}}(1-t)\right).
\end{cases}
\tag{B1}
$$

In order to encourage consistence between center curve of the best candidate and basic path, we first define $E(\boldsymbol{S}_i^{\mathrm{S}})$ as the total difference of displacement terms $\boldsymbol{\delta}$ between $\boldsymbol{C}_i^{\mathrm{S}}$ and $\boldsymbol{C}^{\mathrm{P}}$ in ℓ^2 space. Then by minimizing this term, we estimate the optimal stroke $S^{\mathrm{S}} = \left(\boldsymbol{C}^{\mathrm{S}}, \boldsymbol{L}^{\mathrm{S}}, \boldsymbol{R}^{\mathrm{S}}\right)$ in $\mathcal{C}$ by:

$$
\min_{\boldsymbol{S}_i^{\mathrm{S}} \in \mathcal{C}} E(\boldsymbol{S}_i^{\mathrm{S}}) = \int_0^1 \left\| \boldsymbol{\delta}_i^{\mathrm{S}}(t) - \boldsymbol{\delta}^{\mathrm{P}}(t) \right\|^2 dt,
\tag{B2}
$$

where we use vector calculation defined in Eq. (B3) to evaluate the displacement terms of $\boldsymbol{\delta}$. Intuitively, as shown in Figure B4, for a basic path $\boldsymbol{C}_i^{\mathrm{S}}$, we first define $\hat{\boldsymbol{T}}_i$ as line direction from $\boldsymbol{C}_i^{\mathrm{S}}(0)$ to $\boldsymbol{C}_i^{\mathrm{S}}(1)$, then project the relative position vector $\boldsymbol{C}_i^{\mathrm{S}}(t) - \boldsymbol{C}_i^{\mathrm{S}}(0)$ (green vector) of any point $\boldsymbol{C}_i^{\mathrm{S}}(t)$ on $\boldsymbol{C}_i^{\mathrm{S}}$ to $\hat{\boldsymbol{T}}_i$ (purple vector), finally, compute the displacement $\boldsymbol{\delta}_i^{\mathrm{S}}(t)$ to $\boldsymbol{C}_i^{\mathrm{S}}(t)$ from the nearest point on the line (yellow dashed). Similarly, the displacement $\boldsymbol{\delta}^{\mathrm{P}}(t)$ to $\boldsymbol{C}^{\mathrm{P}}(t)$ over the photo I^{P} can be calculated.

$$
\begin{cases}
\boldsymbol{\delta}_i^{\mathrm{S}}(t) = \boldsymbol{C}_i^{\mathrm{S}}(t) - \boldsymbol{C}_i^{\mathrm{S}}(0) - \boldsymbol{C}_i^{\mathrm{proj}}, \\
\boldsymbol{C}_i^{\mathrm{proj}} = \left(\boldsymbol{C}_i^{\mathrm{S}}(t) - \boldsymbol{C}_i^{\mathrm{S}}(0)\right) \cdot \hat{\boldsymbol{T}}_i, \\
\hat{\boldsymbol{T}}_i = \frac{(\boldsymbol{C}_i^{\mathrm{S}}(1) - \boldsymbol{C}_i^{\mathrm{S}}(0))}{\|\boldsymbol{C}_i^{\mathrm{S}}(1) - \boldsymbol{C}_i^{\mathrm{S}}(0)\|}.
\end{cases}
\tag{B3}
$$

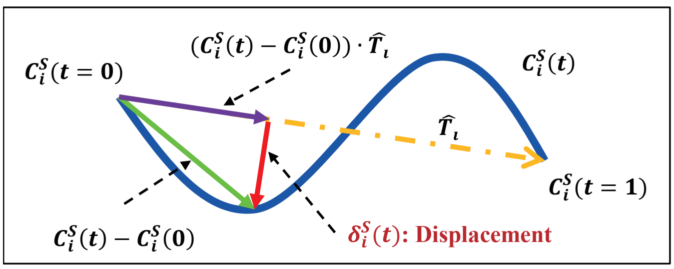


Figure B4 Displacement value for the best stroke candidate.

Appendix B.3 Stroke Generation

After determining the best stroke of style pattern, the basic path sketched by the user will be transformed into an artist stroke with the pattern. This step generates a stroke along the basic path over the given photo by stroke-width determination and texture synthesis.

Aiming to find out major factors that have an influence on variable local widths of a stroke, we have observed shape features of common strokes and their drawing techniques in a large number of artworks of Chinese painting. We ultimately extract the following two key rules: (1) the longer stroke is slenderer; and (2) a stroke widens as it is obviously curved. In other words, stroke width depends on two factors: stroke length and local curvature. Hence, to satisfy two rules, stroke width $w^{\mathrm{P}}(t)$ along the basic path $\boldsymbol{C}^{\mathrm{P}}$ is measured by:

$$w^{\mathrm{P}}(t) = \frac{k_w(t)\kappa^{\mathrm{P}}(t)}{\sqrt{L^{\mathrm{P}}}}, \tag{B4}$$

where $k_w(t)$ is a coefficient to scale the width, and stroke length and curvature are denoted by L^{P} and $\kappa^{\mathrm{P}}(t)$ separately (referring to the length and curvature of C^{P}).

For defining coefficient term $k_w(t)$, we likewise apply the above model to the strokes in given painting. So the width $w^{\mathrm{S}}(t)$ of the optimal stroke in the painting I^{S} satisfies the same form of Eq. (B4), and $k_w(t)$ derives from:

$$k_w(t) = \frac{w^{\mathrm{S}}(t)\sqrt{L^{\mathrm{S}}}}{\kappa^{\mathrm{S}}(t)}, \tag{B5}$$

where $\kappa^{\mathrm{S}}(t)$ is curvature of the path $\boldsymbol{C}^{\mathrm{S}}(t)$, as well as L^{S} refers to the length of $\boldsymbol{C}^{\mathrm{S}}(t)$. Note that we use $\sqrt{L^{\mathrm{P}}}$ in place of L^{P} for weakening the excessive effect caused by a long path.

In terms of stroke width obtained by previous step, two outlines $\boldsymbol{L}^{\mathrm{P}}$ and $\boldsymbol{R}^{\mathrm{P}}$ of a stroke over the given photo can be determined. As shown in Figure B5, we first move feature points on basic path $\boldsymbol{C}^{\mathrm{P}}(t)$ to both sides along their normal directions (yellow arrows) with $w^{\mathrm{P}}(t)$, and then define the domain bounded by $\boldsymbol{L}^{\mathrm{P}}(t)$ and $\boldsymbol{R}^{\mathrm{P}}(t)$ as a fill area of a stroke signed as F^{P}.

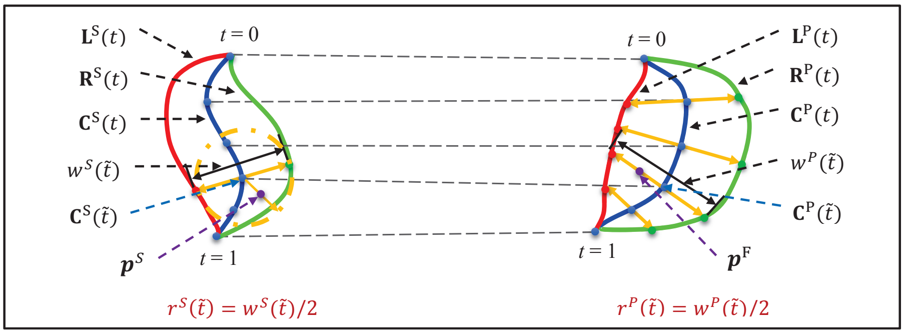


Figure B5 Fill area $\boldsymbol{p}^{\mathrm{F}}$ is shaded by the best pixel $\boldsymbol{p}^{\mathrm{S}}$.

The above process determines local widths of a stroke and marks off a fill area by producing stroke outlines. Next, our goal is to fill the area of a stroke using an approach of texture synthesis based on pixel mapping, where we rasterize the area by scan-line and fill it with the proper pixels selected from the optimal stroke in the painting. In detail, For shading each pixel in a fill area ($\boldsymbol{p}^{\mathrm{F}} \in F^{\mathrm{P}}$) (purple point on the right of Figure B5), which overlaps the pixel in given photo ($\boldsymbol{p}^{\mathrm{P}} \in I^{\mathrm{P}}$), we pick out the best pixel ($\boldsymbol{p}^{\mathrm{S}} \in F^{\mathrm{S}}$) (purple point on the left of Figure B5) from F^{S} of the stroke in I^{S} as the texel, which satisfies the following three constraints: (1) its neighbor pixels have the nearest colour values to the neighborhoods of $\boldsymbol{p}^{\mathrm{F}}$; (2) the relative positions of $\boldsymbol{p}^{\mathrm{F}}$ and $\boldsymbol{p}^{\mathrm{S}}$ to their skeletal paths are the nearest; and (3) the average colour values of the pixels around $\boldsymbol{p}^{\mathrm{S}}$ and $\boldsymbol{p}^{\mathrm{P}}$ are the nearest. We thus define an energy term to describe each of the three constraints.

We define $E_b(\boldsymbol{p}^{\mathrm{S}})$ as the sum of the colour differences of the neighbour pixels between $\boldsymbol{p}^{\mathrm{S}}$ and $\boldsymbol{p}^{\mathrm{F}}$, while $E_a(\boldsymbol{p}^{\mathrm{S}})$ to compute the differences of average colours of the pixels inside circles with a radius R between $\boldsymbol{p}^{\mathrm{S}}$ and $\boldsymbol{p}^{\mathrm{P}}$. We also define $E_p(\boldsymbol{p}^{\mathrm{S}})$ to measure the proximity of $\boldsymbol{p}^{\mathrm{F}}$ and $\boldsymbol{p}^{\mathrm{S}}$ to relevant skeletal paths. Intuitively, the three energy terms correspond to the above constraints respectively.

$$\begin{aligned}
E_b(\boldsymbol{p}^{\mathrm{S}}) &= \textstyle\sum_{i,D\in\{S,F\}}^{\|\boldsymbol{p}_i^{\mathrm{D}}-\boldsymbol{p}^{\mathrm{D}}\|\leqslant 1} \left\|\boldsymbol{\lambda}(\boldsymbol{p}_i^{\mathrm{S}}) - \boldsymbol{\lambda}(\boldsymbol{p}_i^{\mathrm{F}})\right\|, \\
E_p(\boldsymbol{p}^{\mathrm{S}}) &= \left\|\frac{\boldsymbol{p}^{\mathrm{S}} - \boldsymbol{C}^{\mathrm{S}}(\tilde{t})}{r^{\mathrm{S}}(\tilde{t})} - \frac{\boldsymbol{p}^{\mathrm{F}} - \boldsymbol{C}^{\mathrm{P}}(\tilde{t})}{r^{\mathrm{P}}(\tilde{t})}\right\|, \\
E_a(\boldsymbol{p}^{\mathrm{S}}) &= \frac{1}{N}\left\|\textstyle\sum_j^{\|\boldsymbol{p}_j^{\mathrm{S}}-\boldsymbol{p}^{\mathrm{S}}\|\leqslant R} \boldsymbol{\lambda}(\boldsymbol{p}_j^{\mathrm{S}}) - \sum_j^{\|\boldsymbol{p}_j^{\mathrm{P}}-\boldsymbol{p}^{\mathrm{P}}\|\leqslant R} \boldsymbol{\lambda}(\boldsymbol{p}_j^{\mathrm{P}})\right\|,
\end{aligned} \tag{B6}$$

where $\boldsymbol{\lambda}(\boldsymbol{p})$ denotes the color values of $\boldsymbol{p}$, $\boldsymbol{p}_i^{\mathrm{S}}$ is the neighbor pixel of $\boldsymbol{p}^{\mathrm{S}}$, and the same for $\boldsymbol{p}_i^{\mathrm{F}}$ of $\boldsymbol{p}^{\mathrm{F}}$. $\boldsymbol{C}^{\mathrm{P}}(\tilde{t})$ is the nearest point to $\boldsymbol{p}^{\mathrm{F}}$ on the skeletal path and $\boldsymbol{C}^{\mathrm{S}}(\tilde{t})$ refers to the point on the skeletal path of the stroke in I^{S} with the same $\tilde{t}$. As annotated in Figure B5, r^{S} and r^{P} are half of stroke width in our implementation. N is the total number of pixels in a circle of the specified radius R (yellow circle in Figure B5). $\boldsymbol{p}_j^{\mathrm{S}}$ represents the pixels near $\boldsymbol{p}^{\mathrm{S}}$ inside the circle, and the same for $\boldsymbol{p}_j^{\mathrm{P}}$ of $\boldsymbol{p}^{\mathrm{P}}$ in I^{P}.

We combine the three terms together and define the final energy minimization problem in ℓ^1 space as:

$$\min_{\boldsymbol{p}^{\mathrm{S}} \in F^{\mathrm{S}}} [\alpha_1 E_b(\boldsymbol{p}^{\mathrm{S}}) + \alpha_2 E_p(\boldsymbol{p}^{\mathrm{S}}) + \alpha_3 E_a(\boldsymbol{p}^{\mathrm{S}})], \tag{B7}$$

where α_1, α_2, and α_3 are balancing weights (0.3, 0.5 and 0.2 in our experiment). Figure B5 illustrates the process of pixel mapping in our system.

Appendix C Results and Discussion

We have implemented and tested our approach on a large number of photos and paintings with various flower objects using mouse or writing pad. Our technique can automatically convert a basic path over a photo to artistic brush stroke in style of Chinese flower painting through very simple user manipulation (i.e. ambiguous sketch-based selection in Figure C1). Note that our generation is fulfilled at the brush stroke level. Our system has the following important features that make it easy and fast to use.

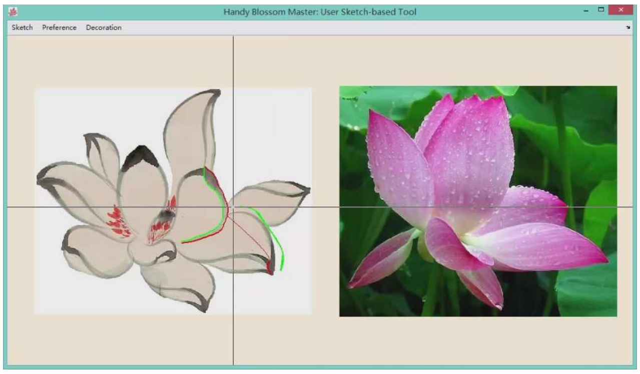


Figure C1 Graphical user interface for roughly sketching the outlines of a stroke on an input painting.

- It just requires imprecise drawing and corrects user-sketched lines to capture user's intention. Thus, given a Chinese painting and a photo with flower target, the user is asked to select stroke candidates as style patterns by roughly sketching the outlines of one or more strokes on the painting, and draw a basic path over the photo.
- It picks out the best stroke from more than one stroke candidates circled by the user for optimum matching with basic path.
- It provides neat and friendly user interface to simplify user manipulation by some rules, for example, the outlines of a stroke are detected whether exist in pairs, final stylistic stroke responds to the operation of path selection in real time, and the user can freely modify the best style pattern yielded by algorithms.
- For preserving the appearance of the style pattern, it automatically calculates local widths that decide stroke outlines, and stylistic texture of an artistic stroke along basic path, once the optimal stroke of style pattern is determined.
- It allows the user to add decorative objects in the output painting, so the user can circle them and paste the entire object directly into the photo, according to a short line drawn by the user on the photo, as a brief guidance of direction, scale and location.

In our experiments, we select flowers-and-birds paintings with high-quality strokes by well-known artists *Zhang-Daqian*, *Qi-Baishi* and *Rao-Zongyi*, which are the most influential and presentative of Chinese paintings, and they are considered difficult to imitate. Four flower sketch paintings with small scale layout are presented in Figure C2. As shown in Figure C4, the paintings in large scale layouts give complete visual effects of Chinese flower paintings with some decorative objects (e.g., leaf, core, stalk etc.), which are picked up directly from paintings and assigned upon user preference. Paintings of other popular objects in Chinese flowers-and-birds painting are displayed, such as chicken, bamboo, morning glory and orchid in Figure C5. Please see the accompanying video for real-time demonstration in the supplementary electronic materials.

Decoration. A complete painting is usually decorated by some objects except the main objects. In flower painting, leaf, core, stalk and grass are used to show the flower to advantage and finish the final painting. We consider and deal with two cases of decorative objects that exist in most of paintings: (1) the object is complete and drawn by thin strokes; and (2) the object is incompletely shown. For the first case, our system allows the user to pick up the objects directly from input painting and assign them with user preference. For the objects in second case, we patch the missing parts on them to fill them up first by means of a method [12]. Figure C3 shows an example of how an incomplete painting image can be filled up following the framework in [12]. Moreover, the user decorates the result in Figure C4 (b) with the leaf in Figure C3 (d).

Limitations. Our algorithm can generate smooth and relatively separate strokes. However, it might not produce satisfactory results in some cases. E.g., some appearances of Chinese artworks are drawn by special strokes with the shape of

Figure C2 Given flower photos captured in various seasons and from different angles of a camera (bottom row), our generated paintings (top bottom) with the stylized strokes in high quality.

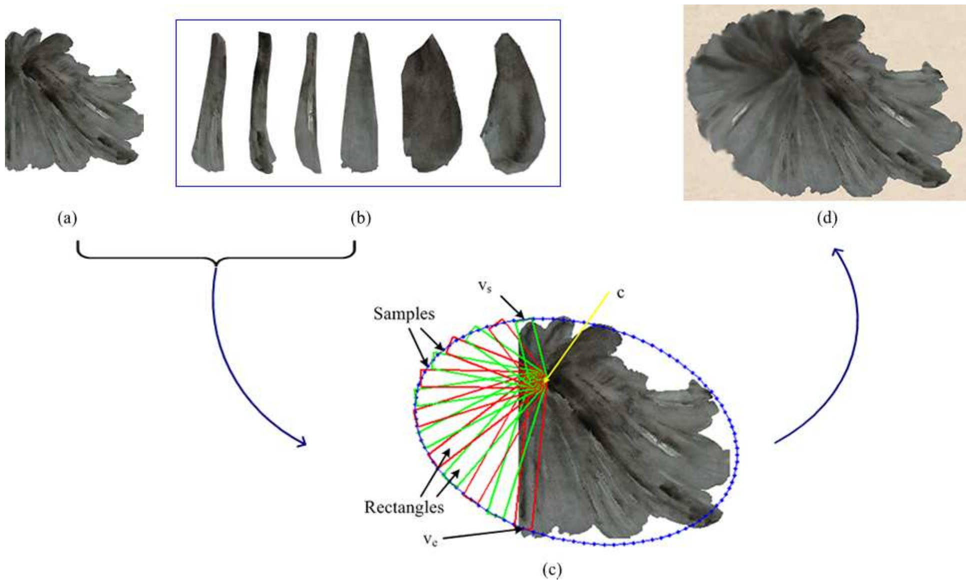


Figure C3 An example of a lotus leaf for decoration. (a) An incomplete painting. (b) The collection of brush stroke texture primitives from (a). (c) The construction of stroke regions. (d) The complete result.

break angle like *wrinkle* in landscape painting. The outlines of these strokes cannot be represented by spline curves. In the case of complicate artwork produced by *splash ink* or multiple overlapped strokes, like western oil painting, multi-layered texture is hardly split due to the mixed colors. But flower painting is a large class of Chinese painting with typical brush strokes, and thus the main case has been handled.

Appendix C.1 User Study

To evaluate the effectiveness of our painting generation algorithm, we have conducted a user study with 12 university students with major in computer science. All of the participants had few opportunities to acquaint Chinese painting and had no background or experience in drawing Chinese flower painting. We selected four well-known flower paintings in different artistic styles by three popular artists: *Zhang-Daqian*, *Qi-Baishi* and *Rao-Zongyi* as the input artworks. At the same time, we offered thirty flower photos to every participant who can arbitrarily pick two photos as the input. The artworks were randomly arranged to the group formed by every three participants (i.e. four groups).

Task. For generating a stylistic flower painting at the level of brush stroke, each participant was asked to use either a manual graphics editor like Adobe Illustrator, or our sketch-based tool. Using the manual editor, the participants first needed to manually accurately sketch stroke outline for extracting every stroke from the painting, and then sketch a path on a photo to be stylized, finally, select a suitable stroke and according to its shape and ink color on the input artwork, draw a stroke along the path. As for our tool, before the study, we let each participant to learn how to use the tool with a brief instructions and practice. In addition, decorative objects (e.g., leaf, core, stalk) were drawn by transforming (i.e. clipping, shifting, scaling and rotation) them from the input artworks. Totally, there are 48 paintings generated by the participants. For each group, 6 paintings by the manual editor and another 6 by our tool.

Performance. We measured the performance of generation tool by the completion time of individual painting, and the usability of each tool. On average, it took 0.64–1.01 minutes per petal to use our tool (more stroke candidates require more time). By comparison, the participants had to averagely spend 9–14 minutes per petal with the manual tool, which is much slower. During the process of user study, we observed that, when using the manual tool, all the participants suffered

(a) (b) (c)

Figure C4 Three completed flower paintings (on the right of (a), (b) and (c)) generated by our system with the input photos (in the upper-left of (a), (b) and (c)) and the masterpiece styles of *Zhang-Daqian*, *Qi-Baishi* and *Rao-Zongyi* (in the lower-left of (a), (b) and (c)), respectively.

difficult to sketch the outlines and fell into a trial-and-error loop to draw the desired stroke, which was extremely time consuming. However, our tool could present stylistic stroke timely as long as the participants roughly sketched the stroke on input artwork and the path on a photo. Simple user interface of our tool and acceptable effects of the produced strokes make painting generation much easier and more efficient.

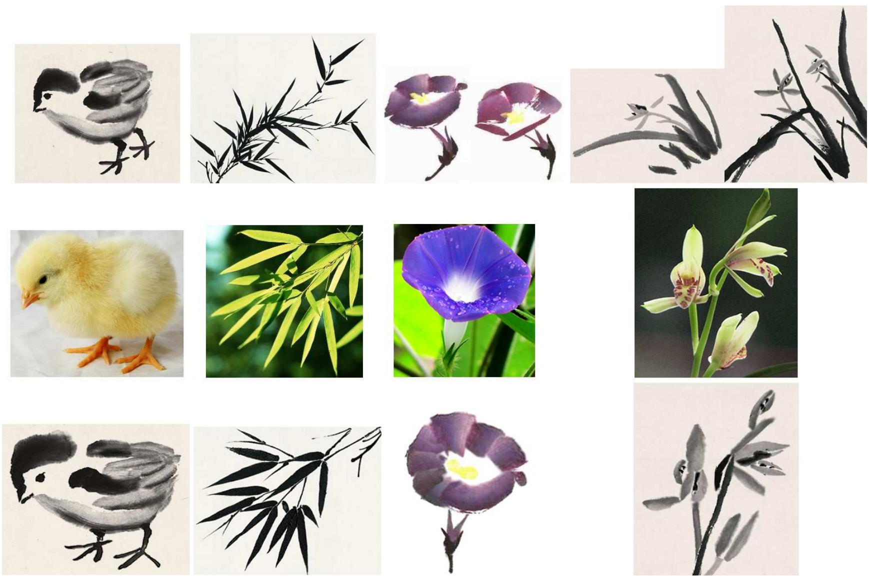


Figure C5 Paintings of other popular objects generated by our system (in the third row) with the input photos (in the second row) and the styles of typical Chinese paintings (in the first row).

Evaluation. To measure the evaluation, three aspects are put forward in the following:

- *Functionality:* whether the user interface is simple, user manipulation is easy, as well as the path on a photo and stroke outlines on the input artworks express user's intent.
- *Visual effect.* whether the style of optimal stroke matches with the path on a photo, and along the path, the outlines and filled texture of the generated stroke are reasonable.
- *Style.* whether the overall appearance of the flower painting is in the similar style to the corresponding input artwork.

In terms of different photo, the above experiment output 24 pairs ($4 \times 3 \times 2$), each of which were produced by the manual tool and our tool respectively. Using paired comparison, we asked the participants involved in painting production to assess the manual tool and our tool in the light of the first two aspects, as well as invited another 6 volunteers who did not join in generation experiment and had background of Chinese flower painting and distributed the pairs equally among them for evaluating the quality of the generated paintings on the third aspect. The assessment of style similarity greatly depends on individual understanding and experience of Chinese painting, which are subjective. Nevertheless, based on the evaluation results quantified by grades from 0 to 9 (higher score means higher appreciation), the results by our tool are scored higher in all aspects of the evaluation (Figure C6). It convincingly proves that our algorithms can generate stylistic brush stroke of Chinese flower painting, which are accepted by the volunteers.

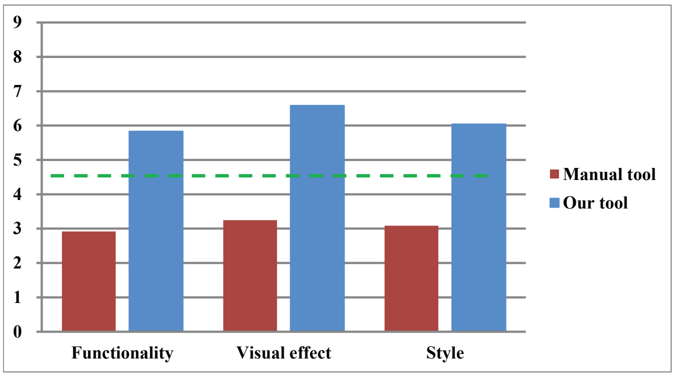


Figure C6 Evaluation results. The vertical axis is the score of the assessment. As the evaluators consider, the data show that our tool significantly outperforms the manual tool on all three aspects ($p < 0.05$).

Appendix D Conclusion and Future Work

This work has presented a handy sketch-based tool for relaxed generation of stylistic brush stroke in Chinese flower painting. The key idea is to explore the smart method of style migration to reduce the complexity and accuracy of user's drawing operation. Our tool performs significantly better than the manual graphics tool in terms of evaluation results in our user study. Our method has the following advantages:

- It can correct and refine rough sketching input including the path over a photo and stroke outlines on an input artwork.
- As the user sketches multiple strokes on the input artwork, it can automatically select the optimal stroke whose style best matches with the path over the photo.
- It can produce the stylistic stroke in real time by mapping the style of shape and ink color onto the basic path, once the path and stroke outlines are drawn imprecisely.

Future work will explore more objects and styles of paintings (e.g., Chinese figure painting and western water color painting) and integrate them with the current framework, which could increase generality of our method. When sketching stroke outlines on an input artwork, our tool still demands the sketched outline must contain one stroke at a time as well as key points for splitting crossing part among several strokes should be specified. This may be labor-intensive particularly there are a large number of strokes on an input artwork. So further reducing the requirements of sketching operation will be another future work.

In summary, we just regard our method as initial attempt in automatic generation of Chinese painting with high-quality brush strokes which involves subjective and intelligent processes. As far as we know, this tool would bring people even children without any background of Chinese artworks closer to Chinese painting.

References

1 Gooch B, Gooch A. In: Non-Photorealistic Rendering. United States: A K Peters, Ltd., 2001.
2 Curtis C J, Anderson S E, Seims J E, et al. Computer-generated watercolor. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM, 1997. 421–430
3 Chu N S H, Tai C L. Moxi: Real-time ink dispersion in absorbent paper. ACM Transactions on Graphics, 2005, 24(3): 504–511
4 Xu S, Tang M, Lau F, et al. Virtual hairy brush for painterly rendering. Graphical Models, 2004, 66(5): 263–302
5 Chu N S H, Baxter W, Wei L Y, et al. Detail-preserving paint modeling for 3D brushes. In: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering. New York: ACM, 2010. 27–34
6 Lu J, Barnes C, DiVerdi S, et al. Realbrush: painting with examples of physical media. ACM Transactions on Graphics, 2013, 32(4): 117:1–117:12
7 Hertzmann A. A survey of stroke-based rendering. IEEE Computer Graphics and Applications, 2003, 23(4): 70–81
8 Hertzmann A, Jacobs C E, Oliver N, et al. Image analogies. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, New York, USA, 2001. 327–340

9 Zhao M, Zhu S C. Customizing painterly rendering styles using stroke processes. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering. New York: ACM, 2011. 137–146
10 Way D L, Shih Z C. The Synthesis of rock textures in Chinese landscape painting. Computer Graphics Forum, 2001, 20(3): 123–131
11 Way D L, Lin Y R, Shih Z C. The synthesis of trees in Chinese landscape painting using silhouette and texture strokes. Journal of WSCG, 2002, 10(2): 499–506
12 Yu J H, Luo G M, Peng Q S. Image-based synthesis of Chinese landscape painting. Journal of Computer Science and Technology, 2003, 18(1): 22–28
13 Xu T C, Yang L J, Wu E H. Stroke-based real-time ink wash painting style rendering for geometric models. In: Proceedings of the ACM SIGGRAPH Asia 2012 Technical Briefs. New York, USA, 2012. 19:1–19:4
14 Yang L J, Xu T C. Animating Chinese ink painting through generating reproducible brush strokes. Sciece China Information Sciences, 2013, 56(1): 1–13
15 Yang L J, Xu T C, Wu E H, et al. Feature-oriented writing process reproduction of Chinese calligraphic artwork. In: Proceedings of the ACM SIGGRAPH Asia 2014 Technical Briefs, New York, USA, 2014. 5:1–5:4
16 Xie N, Laga H, Saito S, et al. IR2s: interactive real photo to sumi-e. In: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering, New York, USA, 2010. 63–71
17 Bang S, Park K. Oriental stylization with strokes and shades. In: Proceedings of the ACM SIGGRAPH '09: Posters, New York, USA, 2009. 87:1–87:1
18 Dong L, Lu S, Jin X. Real-time image-based Chinese ink painting rendering. Multimedia Tools and Applications, 2014, 69(3): 605–620